|  |  |
| --- | --- |
| **FlyweightDemo.java** | |
| **Line** | **Code** |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13**  **14**  **15**  **16**  **17**  **18**  **19**  **20**  **21**  **22**  **23**  **24**  **25**  **26**  **27**  **28**  **29**  **30**  **31**  **32**  **33**  **34**  **35**  **36**  **37**  **38**  **39**  **40**  **41**  **42**  **43**  **44**  **45**  **46**  **47**  **48**  **49**  **50**  **51**  **52**  **53**  **54**  **55**  **56**  **57**  **58**  **59**  **60**  **61**  **62**  **63**  **64**  **65**  **66**  **67**  **68**  **69**  **70**  **71**  **72** | /\*---------------------------------------------------------------------------  This structural code demonstrates the Flyweight pattern  ----------------------------------------------------------------------------\*/  **using** System;  **using** System.Collections.Generic;  **namespace** FlyWeight {  **class** FlyweightFactory {  **private** Dictionary<String, Flyweight> flyweights =  **new** Dictionary<String, Flyweight>();  **public** FlyweightFactory() {  flyweights.Add("X", **new** ConcreteFlyweight());  flyweights.Add("Y", **new** ConcreteFlyweight());  flyweights.Add("Z", **new** ConcreteFlyweight());  }  **public** Flyweight **this**[**string** key] {  **get** {  **if** (!flyweights.ContainsKey(key))  flyweights.Add(key, **new** ConcreteFlyweight());  **return** flyweights[key];  }  }  }  **abstract class** Flyweight {  **public abstract void** Operation(**int** extrinsicstate);  }  **class** ConcreteFlyweight : Flyweight {  **public int** intrinsicState = 10;  **public override void** Operation(**int** extrinsicstate) {  Console.WriteLine("ConcreteFlyweight: " + extrinsicstate);  }  }  **class** UnsharedConcreteFlyweight : Flyweight {  **public override void** Operation(**int** extrinsicstate) {  Console.WriteLine("UnsharedConcreteFlyweight: " + extrinsicstate);  }  }  **class** Program {  **static void** Main() {  //Arbitrary extrinsic state  **int** extrinsicstate = 22;  FlyweightFactory f = **new** FlyweightFactory();  // Work with different flyweight instances  Flyweight fx = f["X"];  fx.Operation(--extrinsicstate);  Flyweight fy = f["Y"];  fy.Operation(--extrinsicstate);  Flyweight fz = f["Z"];  fz.Operation(--extrinsicstate);  //UnsharedConcreteFlyweight uf = new UnsharedConcreteFlyweight();  Flyweight uf = **new** UnsharedConcreteFlyweight();  uf.Operation(--extrinsicstate);  Console.ReadKey();  }  }  }  /\*=================== OUTPUT ===================  ConcreteFlyweight: 21  ConcreteFlyweight: 20  ConcreteFlyweight: 19  UnsharedConcreteFlyweight: 18  =================================================\*/ |